DSP Techniques for Optical Communications

指導教授:吳安宇

報告人:許槐益

Date: 2003/3/5

Forward Error Correction

- System & Algorithm Level
- Architecture Level
- Circuit Level

System & Algorithm Level

Hamming Code

BCH Code

Reed-Solomon Code

Hamming Code

- The well-known (2^n-1, 2^n-n-1) code with Hamming distance of 3 is capable of correcting 1 error
- Error syndrome gives the error location
  - Easy to decode, but only correct one error

BCH Codes

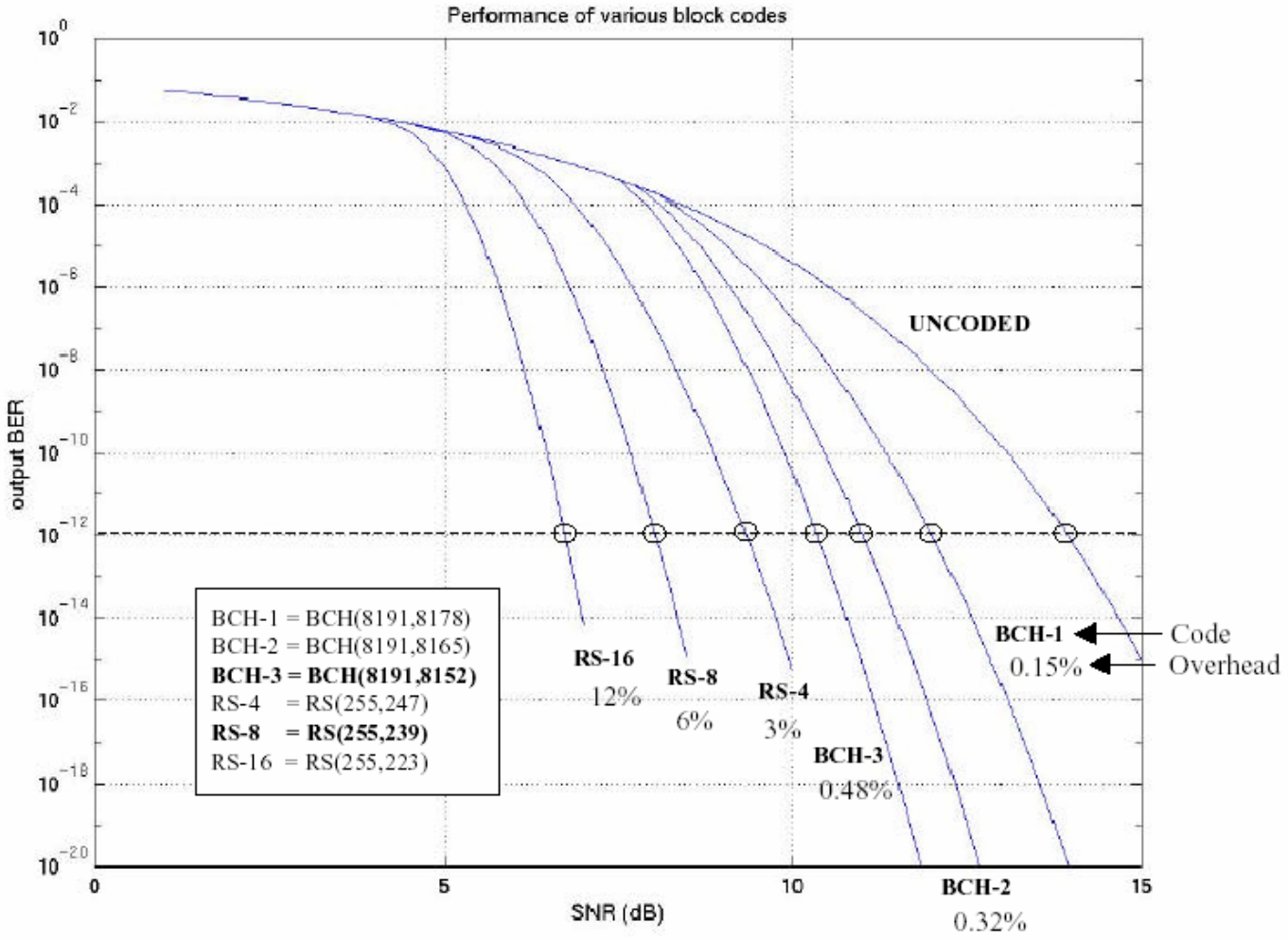
- The cyclic code of length *n* over GF(q), n co-prime with q, whose generator polynomial is g(x).
- Binary BCH code: a BCH code on GF(2), i.e., q=2
- Binary BCH codes are most popular
- BCH codes usually correct few bits in a block
  - 1-error correcting or Hamming
  - 2-error correcting
  - 3-error correcting
- Encoding and decoding procedures are similar to those of RS code, except there is no need to compute error magnitude for binary BCH code

Reed-Solomon Code

- RS(n,k) consists of all polynomials of degree at most n that are multiples of the generator polynomial $g(x)=(x-1)(x-\alpha^{1})\ldots(x-\alpha^{2t-1})$, where α is a primitive element of $GF(2^{m})$

- Minimum distance between codewords is $d=n-k=2t+1$, thus RS(n,k) can correct t symbols
- $2t$ erasures can be corrected (an erasure is an error with known position)

Performance Analysis



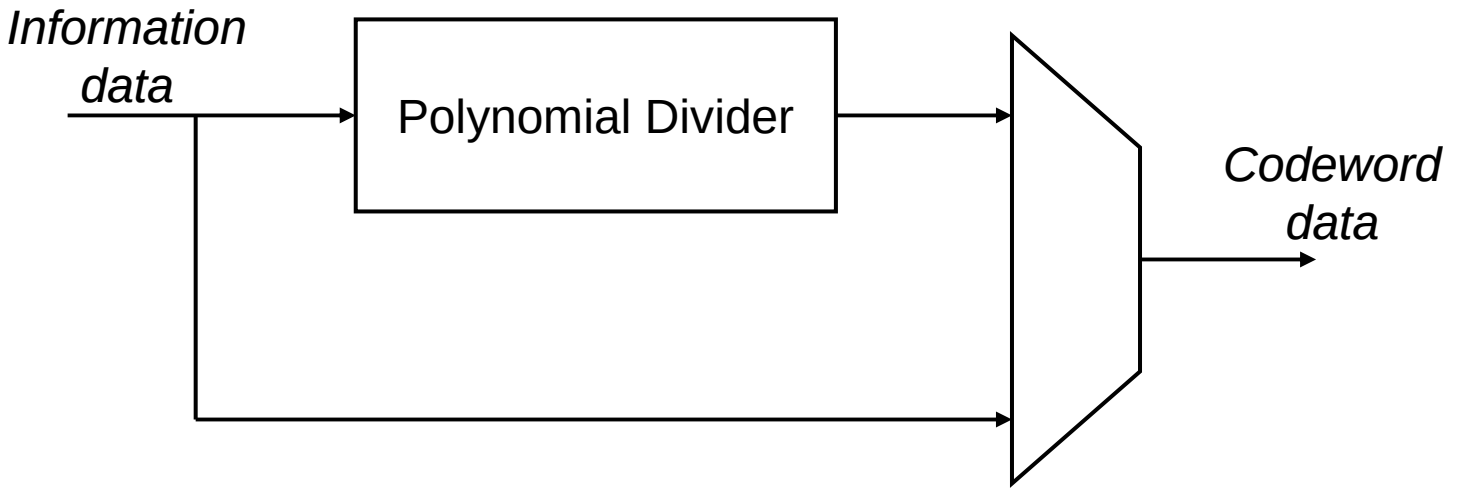
Architecture Level

10Gbps => 3.125Gbps for LX4

3.125Gbps/8 => 390MBps

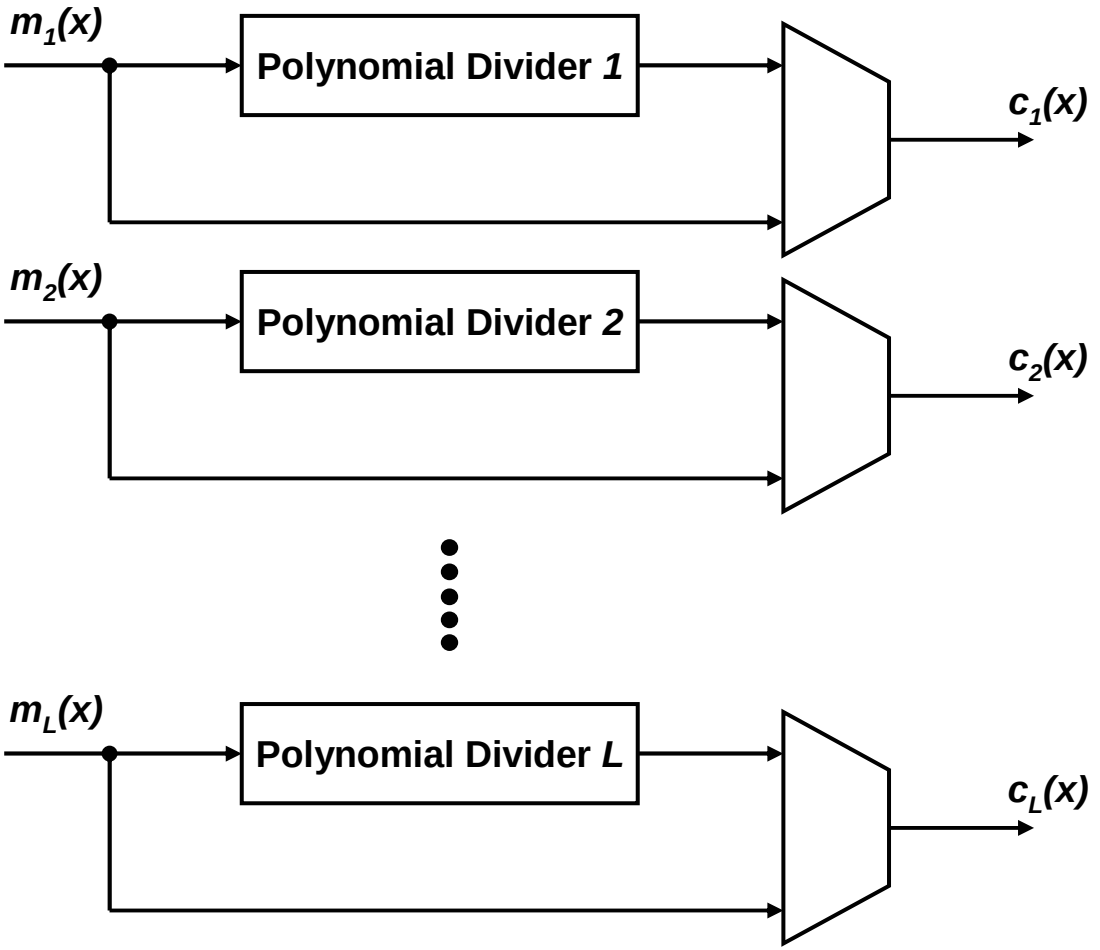
RS Encoder

$$c(x) = x^{2t}v(x) + p(x) = x^{2t}v(x) + x^{2t}v(x) \bmod g(x)$$



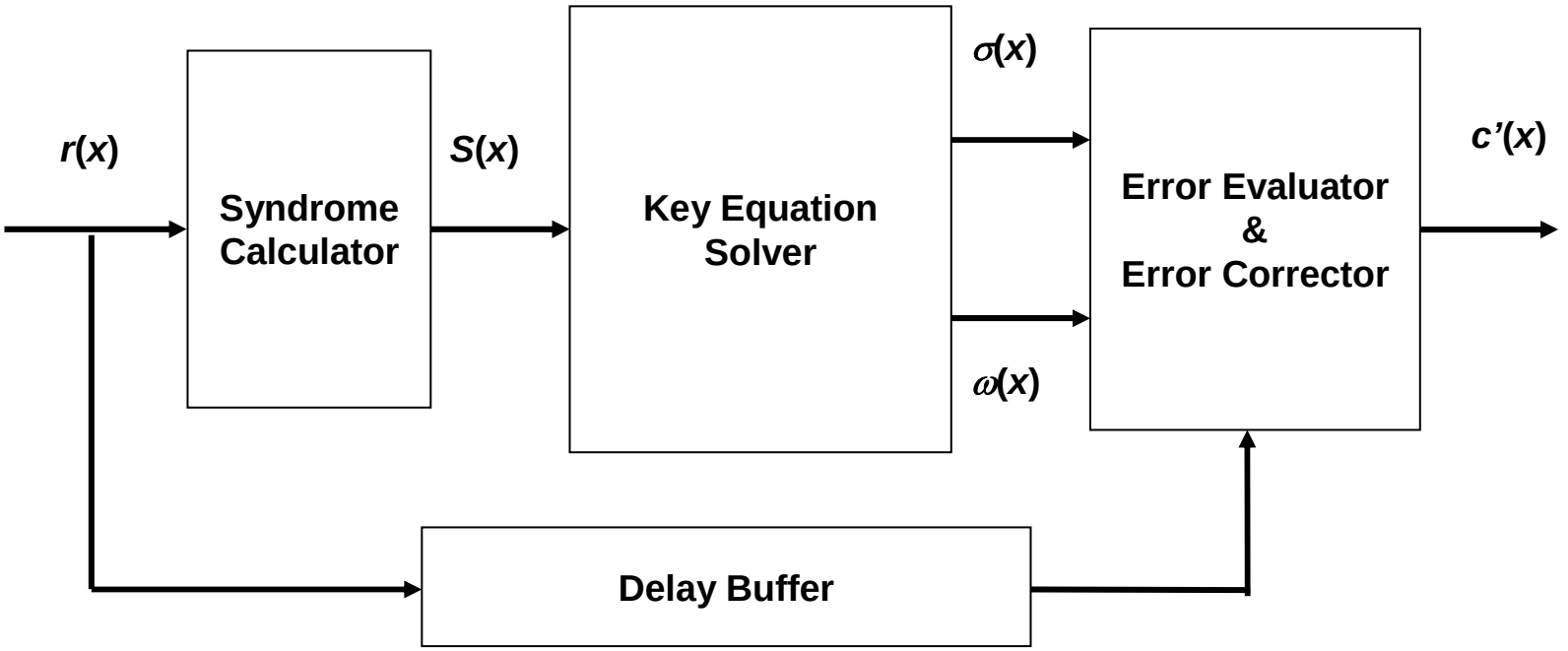
- RS is a cyclic code and often implemented as a LFSR with GF(2^m) operators

Parallel Encoding Architecture



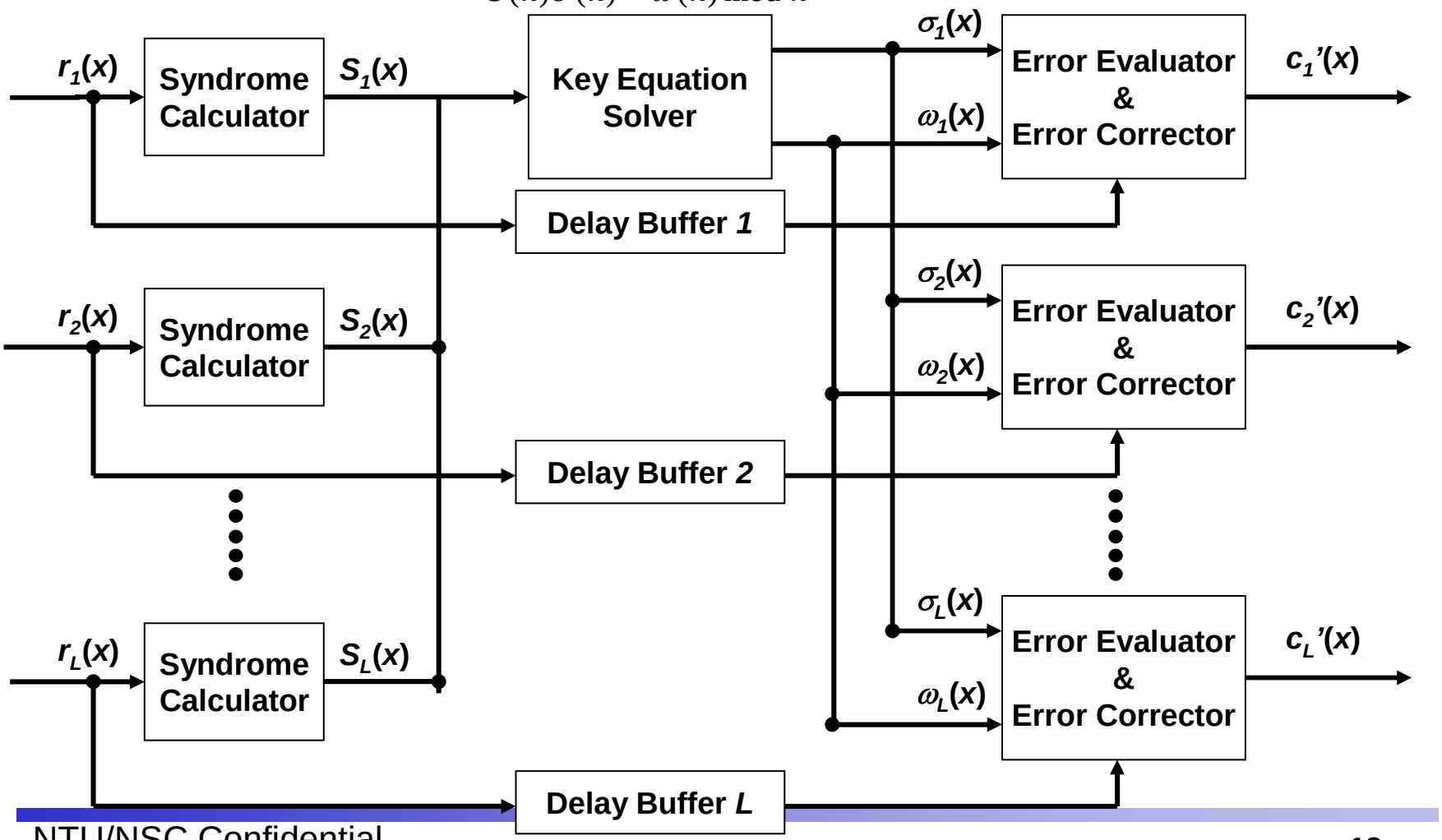
RS Decoder

$$S(x)\sigma(x) = \omega(x) \bmod x^{2t}$$



Parallel Decoding Architecture

$$S(x)\sigma(x) = \omega(x) \bmod x^{2t}$$



Circuit Level

10Gbps => 3.125Gbps for LX4

3.125Gbps/8 => 390MHz => 2.5ns

Optical Communication System

- Forward Error Correcting
  - Data rate: ~10Gbps
  - Operation frequency: 625MHz
  - Clock period: 1.6ns
- Reed-Solomon Codec: RS(255,239)
  - Finite Field Arithmetic
  - Fixed constant coefficients
  - Parallel and Pipeline techniques

Finite Field Arithmetic

- $GF(2^m)=\{0,1,\alpha,\alpha^2,\ldots,\alpha^{2^{m-2}}\}$
- α is the root of a primitive polynomial $p(x)$
- If $m=8$ and $p(x)=1+x^2+x^3+x^4+x^8$
  then α can be represented as (in vector form)
  α = (0100_0000), α^2 = (0010_0000), …, α^8 = (1011_1000), ...
- Three operations:
  - Finite Field Adder (FFA)
  - Finite Field Multiplier (FFM)
  - Finite Field Inversion (FFI)

Finite Field Adder

- Let A,B are two arbitrary symbols (in polynomial form)

$$A = \sum_{i=0}^{m-1} a_i \alpha^i = a_0 + a_1 \alpha^1 + \cdots + a_{m-2} \alpha^{m-2} + a_{m-1} \alpha^{m-1}$$

$$B = \sum_{i=0}^{m-1} b_i \alpha^i = b_0 + b_1 \alpha^1 + \cdots + b_{m-2} \alpha^{m-2} + b_{m-1} \alpha^{m-1}$$

$$a_i, b_i = \{0,1\}$$

$$C = A \oplus B = a_i (xor) b_i \qquad , (0 \le i \le m-1)$$

Finite Field Multiplier

$$C = (A \times B) \bmod p = (A \bmod p) \times B$$

$$C = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-1} \end{bmatrix} = Z(A, p)B = \begin{bmatrix} f_{0,0} & \cdots & f_{0,m-1} \\ \vdots & \ddots & \vdots \\ f_{m-1,0} & \cdots & f_{m-1,m-1} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m-1} \end{bmatrix}$$

where

$$f_{i,j} = \begin{cases} a_i & ; j = 0 \\ u(i-j)a_{i-j} + \sum_{t=0}^{j-1} p_{j-1-t,i}\, a_{n-t-1} & ; j = 1 \sim n-1 \end{cases} \quad ; j = 0 \sim n-1$$

$$u(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \qquad q_{i,j} = \begin{cases} q_{i-1,n-1} & ; j = 0 \\ q_{i-1,j-1} + q_{i-1,n-1} q_{0,j} & ; j = 1 \sim n-1 \end{cases} \quad ; i = 1 \sim n-2$$

Finite Field Multiplier

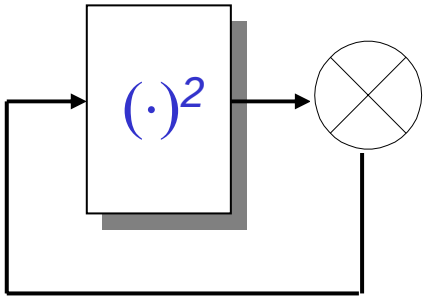
- If $m=8$ and $p(x)=1+x^2+x^3+x^4+x^8$, then

$$Z=\begin{bmatrix}
a_0 & a_7 & a_6 & a_5 & a_4 & a_3+a_7 & a_2+a_6+a_7 & a_1+a_5+a_6+a_7 \\
a_1 & a_0 & a_7 & a_6 & a_5 & a_4 & a_3+a_7 & a_2+a_6+a_7 \\
a_2 & a_1+a_7 & a_0+a_6 & a_5+a_7 & a_4+a_6 & a_3+a_5+a_7 & a_2+a_4+a_6+a_7 & a_1+a_3+a_5+a_6 \\
a_3 & a_2+a_7 & a_1+a_6+a_7 & a_0+a_5+a_6 & a_4+a_5+a_7 & a_3+a_4+a_6+a_7 & a_2+a_3+a_5+a_6 & a_1+a_2+a_4+a_5 \\
a_4 & a_3+a_7 & a_2+a_6+a_7 & a_1+a_5+a_6+a_7 & a_0+a_4+a_5+a_6 & a_3+a_4+a_5 & a_2+a_3+a_4 & a_1+a_2+a_3+a_7 \\
a_5 & a_4 & a_3+a_7 & a_2+a_6+a_7 & a_1+a_5+a_6+a_7 & a_0+a_4+a_5+a_6 & a_3+a_4+a_5 & a_2+a_3+a_4 \\
a_6 & a_5 & a_4 & a_3+a_7 & a_2+a_6+a_7 & a_1+a_5+a_6+a_7 & a_0+a_4+a_5+a_6 & a_3+a_4+a_5 \\
a_7 & a_6 & a_5 & a_4 & a_3+a_7 & a_2+a_6+a_7 & a_1+a_5+a_6+a_7 & a_0+a_4+a_5+a_6
\end{bmatrix}$$

- When A is a constant, it requires less silicon area

Finite Field Inversion

$$\frac{1}{\alpha^1} = \frac{\alpha^{2^m-1}}{\alpha^1} = \alpha^{2^m-2} = \alpha^2\alpha^{2^2}\cdots\alpha^{2^{m-1}}$$



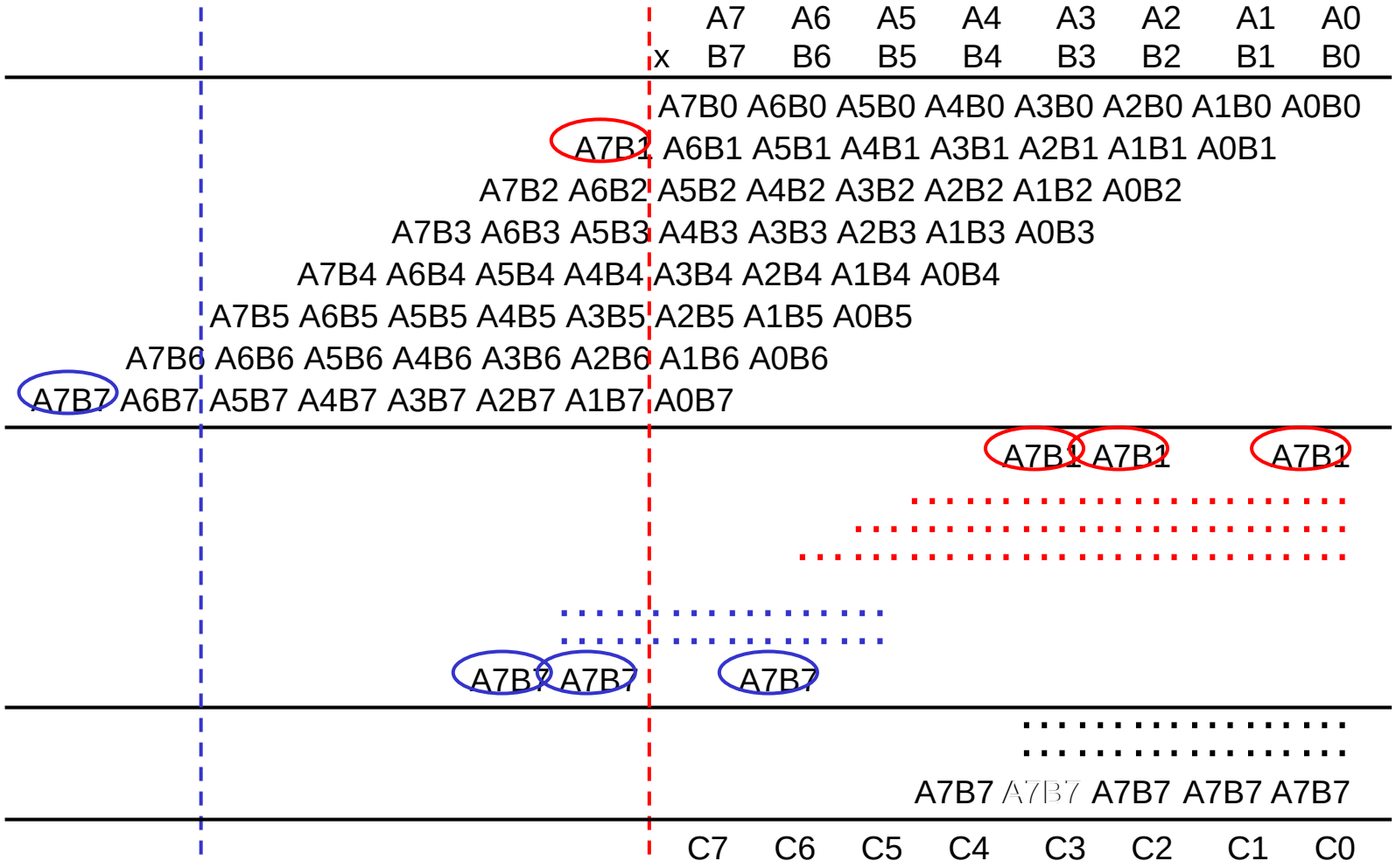
m iteration

- Need squaring and multiplier
- Can be designed using combinational logic
- Only used one in RS decoding process

Finite Field Arithmetic GF(2^8)

- Finite Field Addition (FFA)
  - 24 gates // 8 cells
  - 0.37 ns
- Finite Field Multiplication (FFM)
  - 287 gates // 158 cells
  - 4.11 ns
- Finite Field Inversion (FFI)
  - 661 gates // 537 cells
  - 4.84 ns

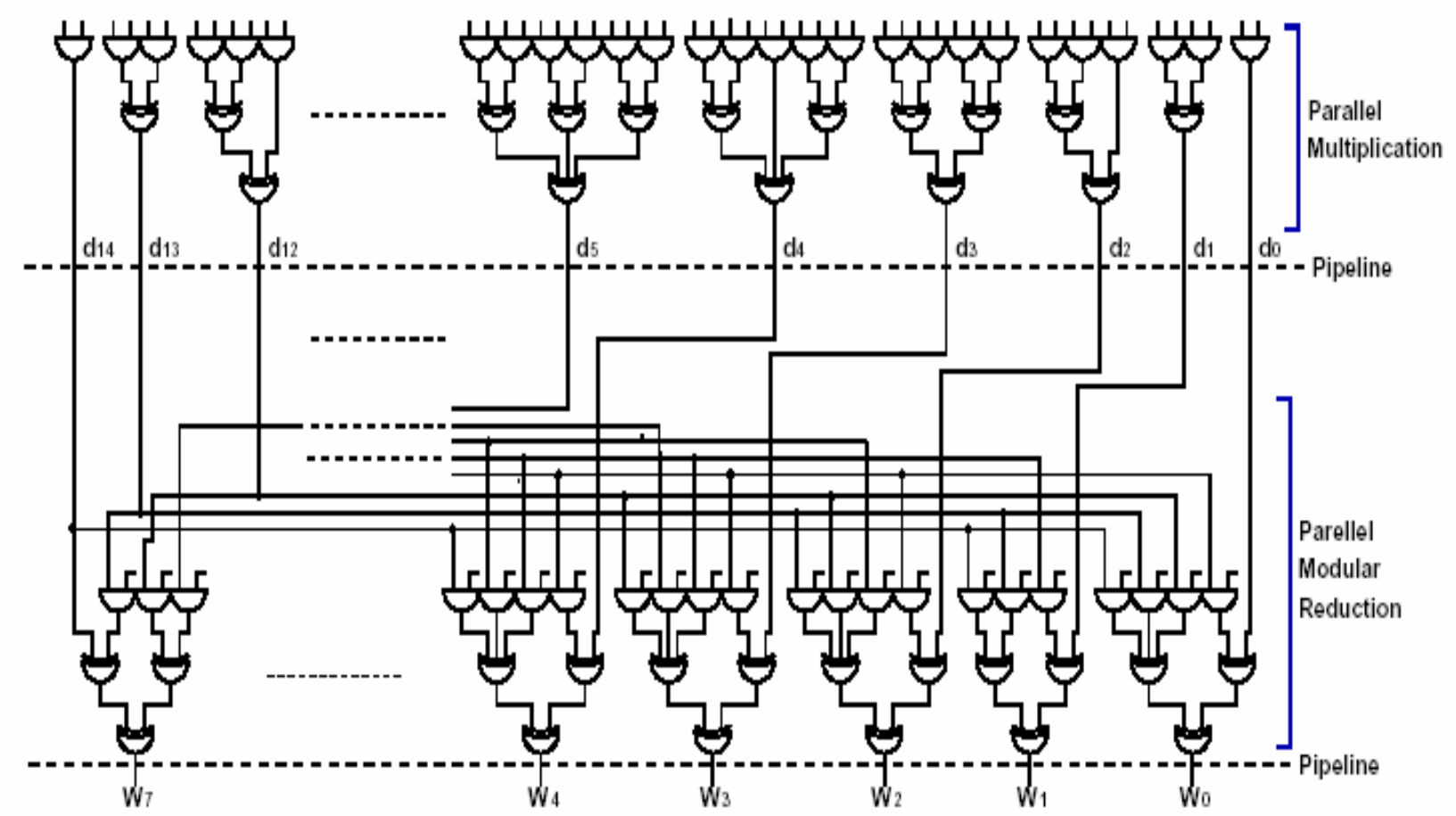
Array FFM Implementation



Pipeline FFM

- Search similar term
  - X17, X27, X37, X57, X06, X46,
  - X12, X23, X34, X45, X56, X13.
- Pipeline1
  - First Stage: $3T_{XOR}$ (12XOR+15XOR)
  - Finial Stage: $1T_{AND} + 3T_{XOR}$ (64AND+56XOR)
- Pipeline2
  - First Stage: $3T_{XOR} + 1T_{AND}$ (12XOR+15XOR+64AND)
  - Finial Stage: $3T_{XOR}$ (56XOR)

A pipelined fully-parallel multiplier over GF(2^8)



Conclusions

- FEC greatly improve BER for given SNR
- FEC Design Issues
  - RS(255,239)+RS(255,239)/BCH(4359,4320)
  - High speed VLSI architecture
  - Low cost
  - Low power

0.25um Cell Library

Logic Symbol

Cell Size

Drive Strength	Height (μm)	Width (μm)
XOR2XL	6.4	7.2
XOR2X1	6.4	8.1
XOR2X2	6.4	9.0
XOR2X4	6.4	16.2

46.08

103.68

Delays at 25°C, 2.5V, Typical Process

Description	Intrinsic Delay (ns)			
	XL	X1	X2	X4
A → Y↑	0.229	0.209	0.197	0.153
A → Y↓	0.238	0.194	0.167	0.156
B → Y↑	0.329	0.189	0.162	0.161
B → Y↓	0.346	0.216	0.186	0.184

Logic Symbol

Cell Size

Drive Strength	Height (μm)	Width (μm)
AND2XL	6.4	3.6
AND2X1	6.4	3.6
AND2X2	6.4	3.6
AND2X4	6.4	6.3

23.04

40.32

Delays at 25°C, 2.5V, Typical Process

Description	Intrinsic Delay (ns)			
	XL	X1	X2	X4
A → Y↑	0.114	0.120	0.095	0.086
A → Y↓	0.158	0.179	0.141	0.125
B → Y↑	0.122	0.127	0.101	0.091
B → Y↓	0.183	0.203	0.162	0.145